



Universidad  
Carlos III de Madrid



This is a postprint version of the following published document:

Frutos-López, M., González-de-Suso, J. L., Sanz-Rodríguez, S., Peláez-Moreno, C. & Díaz-de-María, F. (2015). Two-level sliding-window VBR control algorithm for video on demand streaming. *Signal Processing: Image Communication*, 36, 1-13. DOI: <http://dx.doi.org/10.1016/j.image.2015.05.004>

© 2015 Elsevier B.V.



**This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.**

# Two-Level Sliding-Window VBR Control Algorithm for Video on Demand Streaming

Manuel de-Frutos-López<sup>a</sup>, José Luis González-de-Suso<sup>b</sup>, Sergio Sanz-Rodríguez<sup>c,\*</sup>, Carmen Peláez-Moreno<sup>b</sup>, Fernando Díaz-de-María<sup>b</sup>

<sup>a</sup>*Audio und Multimedia Echtzeitssysteme, Fraunhofer IIS, Erlangen, Germany*

<sup>b</sup>*Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Leganés, Spain*

<sup>c</sup>*Embedded Systems Architecture, Technische Universität Berlin, Berlin, Germany*

---

## Abstract

A two-level *variable bit rate* (VBR) control algorithm for hierarchical video coding, specifically tailored for the new High Efficiency Video Coding (HEVC) standard, is presented here. A long-term level monitors the current bit count along a sliding window of a few seconds, comprising several *intra periods* (IPs) and shifted on an IP basis. This long-term view allows the accommodation of the naturally occurring rate variations at a slow pace, avoiding the annoying sharp quality changes commonly appearing when non-sliding window approaches are used. The bit excesses or defects observed at this level are evenly delivered to a short-term level mechanism that establishes target bit budgets for a narrower sliding window covering a single IP and shifting on a frame basis. At this level, an adequate quantization parameter is estimated to comply with the designated target bit rate.

Recommended test conditions as well as two few minutes long video sequences with scene cuts have been used for the assessment of the proposed VBR controller. Comparisons with a state-of-the-art rate control algorithm have produced good results in terms of quality consistency, in exchange for moderate rate-distortion performance losses.

*Keywords:* Video coding, HEVC, rate control, VBR, sliding window, video storage, video on demand.

---

## 1. Introduction

The increasing popularity of video services and entertainment in the current Information Age has greatly been prompted by the continuous advances in wireless transmission networks, Internet, data storage systems, *high definition* (HD) and 3D video, as well as by important improvements in memory and computational capabilities of multimedia devices.

Video compression standards have also played a paramount role in the success of video applications. MPEG-2 [1], H.263 [2], MPEG-4 Visual [3] and H.264/Advanced Video Coding (AVC) [4] are some of the most widely employed video coding standards that were developed to meet the demands of those times. Nowadays, the wide variety of existing video applications, the growing demand on HD video exchange, as well as the arrival of the 4K resolution have made the previous standards obsolete calling for the next generation, capable of achieving high compression efficiency, not only for these resolutions, but also for the traditional ones, typically requested by mobile and broadcasting applications. This need is expected to be fulfilled by the new High Efficiency Video Coding (HEVC) standard [5].

---

\*Corresponding author. Present address: Technische Universität Berlin - Institut für Technische Informatik und Mikroelektronik (TIME). Einsteinufer 17, D-10587 Berlin, Germany. Tel.: +49 30 31421-357. Fax: +49 30 31422-943.

*Email addresses:* [manuel.frutos@iis.fraunhofer.de](mailto:manuel.frutos@iis.fraunhofer.de) (Manuel de-Frutos-López), [jlmolinero@tsc.uc3m.es](mailto:jlmolinero@tsc.uc3m.es) (José Luis González-de-Suso), [sergio.sanz@aes.tu-berlin.de](mailto:sergio.sanz@aes.tu-berlin.de) (Sergio Sanz-Rodríguez), [carmen@tsc.uc3m.es](mailto:carmen@tsc.uc3m.es) (Carmen Peláez-Moreno), [fdiaz@tsc.uc3m.es](mailto:fdiaz@tsc.uc3m.es) (Fernando Díaz-de-María)

To this end, HEVC incorporates novel features in the *video coding layer* (VCL) aimed at improving compression efficiency, advances in the *network abstraction layer* (NAL) focused on providing a more flexible and robust video transmission, as well as some parallel-friendly coding strategies to enable real-time processing especially in those HD and beyond-HD resolutions (for further details about HEVC, the reader is referred to [6]).

Concerning the encoding pattern and similarly to the H.264 standard, the pictures belonging to a *group of pictures* (GoP) in HEVC can be organized into temporal layers, with the length of a GoP defined as the distance between two consecutive pictures from the temporal base layer. As depicted in Fig. 1, the pictures of this base layer can be *intra* (I)-predicted or inter-predicted, the latter by means of prediction with respect to an already encoded picture or bi-prediction (B) with respect to one or two encoded pictures; whereas the pictures of an enhancement layer can be inter-predicted from references belonging to lower layers. Specifically, in this so-called hierarchical coding two kinds of strategies can be distinguished: dyadic hierarchical B-picture coding and zero-delay hierarchical P-picture or B-picture coding. The former imposes a one-GoP algorithmic delay in order to take advantage of a B-type motion-compensated prediction with one past and one future reference frame. As can be seen in Fig. 1, those frames belonging to the highest temporal layer can be predicted from nearer references and, therefore, will produce fewer bits when encoded, while the prediction in lower layers is not so accurate due to the increasing distance to their references. This kind of hierarchical coding is generally used for those applications demanding high quality of reconstructed video such as digital storage and *video on demand* (VoD) streaming, the ones considered in this paper. The latter, on the other hand, does not need any picture disordering for motion-compensated prediction, since it is always accomplished from past references, thus being a coding pattern suitable for low-delay applications and out of the scope of this paper. In H.264 these hierarchical GoP structures have been shown to improve the *rate-distortion* (R-D) performance compared to classical coding patterns [7, 8], and this benefit has been efficiently exploited in HEVC, where new coding tools have been developed taking into account this kind of patterns.

Although high compression efficiency is generally pursued in video coding systems, most of multimedia applications demand such a requirement under some bit rate constraints. For this reason, a *rate control algorithm* (RCA) must be embedded in the video encoder, which it is responsible for adjusting the output bit rate under certain buffer constraints given a specified target bit rate, while maximizing the compressed video quality. To this end, the RCA must be able to properly regulate some encoder parameters affecting the bit rate, typically the *quantization parameter* (QP) of transform coefficients.

In general, two kinds of RC strategies for video coding can be distinguished: *constant bit rate* (CBR) and *variable bit rate* (VBR) controllers. On the one hand, CBR controllers, generally used for real-time low-delay transmission applications, produce short-term bit rate adjustments to meet low buffer delay constraints. Thus, a fine control of the instantaneous bit rate around a *fixed reference point*, the target bit rate, is pursued at the expense of large variations in quality due to the video content variability in natural scenes. On the other hand, VBR controllers, typically employed for video storage and streaming applications, pursue a long-term bit rate adaptation to provide smoother quality fluctuations at the expense of a longer buffer delay needed to accommodate potentially higher instantaneous bit rate variations [9, 10]. Since in VBR coding the instantaneous output bit rate is not the biggest concern as long as the average output bit rate requirements are met, more resources in the most complex scenes can be allocated, thus resulting in a potential overall R-D benefit and good quality consistency. Consequently, VBR controllers in combination with dyadic hierarchical GoP structures are the preferred coding solution when the bit rate constraints are, to some extent, relaxed.

In this paper, a two-layer VBR controller for hierarchical video coding is proposed. The long-term layer allocates bits in a large window of a few seconds according to the deviations detected with respect to the target average bit rate, and a short-term layer changes the frame QP smoothly in order to approximately achieve the target bit budget in a shorter window. While the long-term layer consists in an evolution of the bit bucket mechanism, the short-term layer makes use of predictors for anticipating bit rate deviations and to decide whether or not is it necessary to change the QP for the actual frame. In this way, the complexity fluctuations of the different parts of the sequence are not translated into high quality fluctuations. The use of sliding analysis windows in both long and short-term layers guarantees a smooth response of the algorithm,

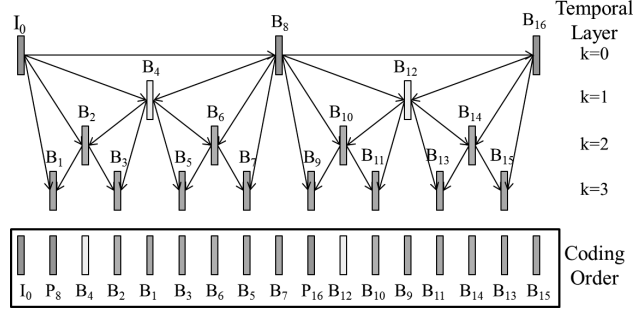


Figure 1: Dyadic hierarchical coding for a GoP size of 8 frames. Notice that the inter frames in layer  $k=0$  are labeled as B, since in the more general case they can be bi-predicted. In practice, they are often encoded using single prediction.

essential in VBR environments.

In summary, the proposed algorithm is simple, effective, and easily adaptable to a variety of scenarios by properly dimensioning the span of the long-term window, in this paper specifically configured for video storage or VoD applications. Furthermore, the proposed VBR controller has been implemented and tested on the HEVC standard, for which, as far as we know, there are still no specific solutions.

This paper is organized as follows: Section 2 presents the general theory and state of the art in RC, specifically addressing VBR scenarios and their constraints; Section 3 details the proposed VBR controller; the experimental set up for assessing its performance follows in Section 4 ending in Section 5 with our conclusions and an outline for future research.

## 2. Rate Control in VBR Environments

In this section, the need of an RCA in video coding applications, its functionality under certain bit rate constrains, as well as the state of the art in RC are described, to end up with a discussion focused on the specific constraints of VBR applications over the output of a video encoder and their influence on the coding performance.

### 2.1. Quantization Parameter Cascading

Leaving aside low-delay applications, as we explained in Section 1, and focusing on the use of hierarchical GoP structures, a key point to be considered is the way in which a QP value is associated to each of the temporal layers. In terms of R-D efficiency, the pictures of the lowest temporal layer ( $k = 0$  in Fig. 1) should be encoded with the highest quality since they are the ones used as references for motion-compensated prediction of pictures belonging to higher layers. The pictures of the subsequent layer should be encoded with lower quality, since fewer pictures will be referred to them, and so on for deeper layers. In short, QP should increase progressively from one temporal layer to the next. Although this *QP cascading* (QPC) approach may incur in certain distortion fluctuations among temporal layers, the subjective quality of reconstructed video is not perceived as annoying [7], in general.

Though several alternatives for cascading the QP over temporal layers have been proposed for H.264/AVC [8, 7, 11] and HEVC [12, 13], they are not capable of dynamically adapting the QP. To meet some commonly required constraints such as a particular transmission buffer or a maximum bit budget for storage they have to be complemented with an RCA.

### 2.2. Rate Control Functionality: Buffer Design

The output bit rate of a video encoder, even when the coding parameters remain constant, may exhibit important temporal variations. These variations have to do with changes in video content, the use of different picture coding types (I, P or B) as well as different coding options as the aforementioned QPC, for

example. As described in [10], and with the exception of those channels capable of delivering data without any restriction to the bit rate, this means that an output buffer or leaky bucket is always placed as an interface between the encoder and the transmission channel. Also, at the decoder side and before decoding a particular frame, all the bits corresponding to that frame need to be stored in a decoder buffer. As a consequence, there are mainly two situations that a RCA must control: *underflow*, when the decoder buffer has not received the whole frame information on time (perceived as a discontinuity in the video display); and *overflow*, when the channel bit rate and the sizes of the encoded pictures are such that the decoder buffer capacity is surpassed and, therefore, information of some pictures is lost (perceived as jumps in the sequence when displaying).

Decoder buffer overflow is not an issue in packet switched networks, since the channel can be switched off when there is no data to transmit. Underflow can be prevented by choosing an initial decoding delay large enough to enable the buffering of a certain amount of data before starting to decode pictures. Although these techniques are widely employed in some streaming applications, they produce an important delay that should be avoided in others with stronger delay constraints like, for example, delay-sensitive streaming, live video broadcast, or videoconferencing, where the only solution is to strictly monitor and control the output bit rate of the encoder.

### 2.3. Rate Control Parameters for VBR Environments

To set up the problem of designing a VBR-RCA considering the additional degree of freedom with respect to a CBR-RCA, we define the following parameters with the aid of Fig. 2 (taken from [14]):

- **Target bit rate ( $R_T$ )**: The average bit rate for the whole video sequence measured in bits per second (bps). In storage applications, it is more usual to employ the target bit count in bits,  $B_T$ , as showed in Fig. 2. If  $N_F$  is the total number of pictures in the sequence, both parameters are related by:  $B_T = N_F f^{-1} R_T$ , where  $f$  is the number of frames per second.
- **Maximum bit rate ( $R_M$ )**: For storage applications it can be identified as the highest access speed of the storage device. Fig. 2 represents  $R_M$  as the maximum instantaneous increase (slope) allowed in the bit count curve. For streaming applications  $R_M$  is generally bounded by the channel bit rate.
- **Maximum exceeded bit count (MEBC)**: For storage applications this is the number of bits that the encoded sequence can exceed the initially targeted bit count, expressed as a percentage of  $B_T$ . In order to use a similar constraint for those streaming applications in which the length of the sequence is unknown, the so-called *instantaneous MEBC*, or  $MEBC_I$ , can be defined as a bound to the bit count (or to the average bit rate) at each time instant. Its value, represented as a blue dotted line of Fig. 2, increases with the length of the sequence. Occasionally exceeding  $MEBC_I$ , however, is not critical as long as MEBC is not surpassed at the end of the sequence.

### 2.4. State of the Art

A wide amount of RCAs have been proposed in the literature, especially during the emergence of each new video coding standard in order to meet the demands of the most popular applications of the moment. Many of these algorithms provide practical solutions for CBR environments that meet short-term bit rate constraints by means of analytical *rate-quantization* (R-Q) models for accurate QP estimation. The same models have also been adopted by some RCAs in VBR scenarios. However, since VBR controllers do not need to meet strict short-term rate constraints (as already pointed out in Section 1), another successful VBR approach has been proposed that aims at providing improved quality consistency subject to a long-term rate constraint by producing smooth QP variations according to the actual rate and distortion produced in already encoded frames. For a review of the most relevant CBR and VBR controllers designed for video coding standards prior to HEVC the reader is referred to [15, 16].

Several RCAs have been presented for the HEVC standard with hierarchical GoP structures. Most of them are inspired in some well-known CBR-RCAs previously proposed for H.264/AVC: Naccari *et al.* [17] and Choi *et al.* [18] proposed CBR controllers based on that described in [19], which employed the commonly

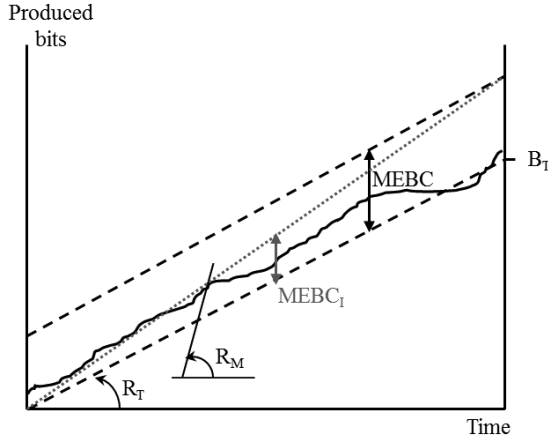


Figure 2: Bit count curve representing the bit rate constraints for VBR scenarios.

used quadratic R-Q model for QP estimation; Si *et al.* [20] proposed an RCA based on the simple linear R-Q model stated in [21]; Wang *et al.* [22] used for their RCA the well-known  $\rho$ -domain model described in [23]; Yoon *et al.* [24], Sanz *et al.* [25], Liang *et al.* [26] and Sun *et al.* [27] presented RCAs that use the accurate Cauchy-density-based R-Q model already proposed in [28]. This latter paper uses specifically a pre-analysis step to predict accurately the source rate.

Alternatively, assuming that the bit rate is more sensitive to the *Lagrange multiplier* ( $\lambda$ ) for R-D optimization than the QP [12], Li *et al.* [29] proposed a novel RCA consisting of computing first, from the bit budget targeted, an appropriate  $\lambda$  value using the Cauchy-density-based R-D function for transform coefficient modeling [28] and then, the QP using a simple  $\lambda$ -QP mapping function. Owing to its excellent R-D performance, this algorithm has been adopted by the Joint Collaborative Team on Video Coding (JCTVC) as the default RCA in the test model (HM) reference software [13], thus being a benchmark for comparison purposes. However, it also deserves some critical comment regarding its suitability in a VBR scenario: the algorithm in [29] is not able to smoothly compensate the observed bit rate deviations from the target bit rate, since the bit resources allocation is similar to that of a regular CBR-RCA. In other words, no long-term target bit rate adaptation is pursued, thus affecting somehow the quality consistency.

Other alternative QP adjustment models already tested on H.264/AVC video coding for VBR applications, such as those that produce smooth QP variations relying on the actual rate and distortion of previously encoded frames [16, 30], have not been applied yet to HEVC. This kind of QP control strategy is followed in this paper.

It is worth mentioning that all of the RCAs proposed for the new video coding standard use QPC for coding efficiency maximization as we described in Section 2.1.

### 3. Proposed Two-Level Sliding-Window VBR Controller

Since we target the VBR strategy, the aim of our proposal is to control the average output bit rate of the encoder on a *long-term* basis by modifying the QP employed at frame level. Without exceeding the VBR constraints, the instantaneous output bit rate is modulated to avoid large quality variations by means of a robust two-level sliding-window strategy for bit resource allocation that is, subsequently, translated into smooth QP variations, thus resulting in a good quality consistency.

The block diagram in Fig. 3 illustrates the operation of the proposed two-level VBR controller. First, the mismatch between the expected and the produced amount of bits is monitored by the *long-term* (LT) layer by using an analysis window of a few seconds that is composed of an integer number  $N$  of *intra periods* (IPs). An acceptable bit count range is defined, taking into account the MEBC imposed by the application

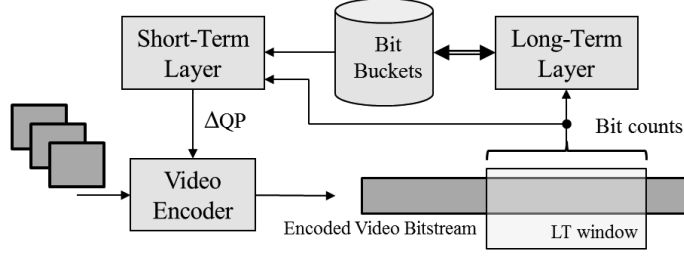


Figure 3: Block diagram of the proposed two-level VBR controller.

constraints. According to the deviation observed in the current LT window with respect to the limits of this acceptance range, the bit budget available for encoding the following  $N$  IPs is modified. In particular, a set of offset values, one for each of the following  $N$  IPs, is stored in the so-called array of *bit buckets*.

As will be described later on, the target bit rate for each IP is modified by the value stored in its corresponding bit bucket. In this way, the bit allocation scheme smoothly compensates the differences with respect to the target average bit rate employing the available bits more wisely in those regions of the sequence that exhibit a higher complexity.

In the low level, a *short-term* (ST) layer control is imposed, in order to accomplish the target bit budget previously allocated by the LT-layer for a ST sliding window of  $M$  pictures. In our proposal  $M$  is fixed to the number of frames in an IP, in order to ensure that each ST-window spans exactly the same number of frames of each type and hierarchical level. An estimation of the bit count corresponding to the current ST window is then compared to its bit budget in order to determine a proper QP to compensate for the expected bit rate deviations. The operation of both LT and ST layers is described in detail in the following subsections.

### 3.1. Long-Term Layer

As depicted in Fig. 4, each LT analysis window is designed to cover an integer number  $N$  of IPs shifting on an IP basis; thus, the  $i$ th window ends with the  $i$ th IP and there is an overlap of  $(N-1)$  IPs between two consecutive LT windows.

This fact, together with a wise distribution of the differences in bit occupation among  $N$  different IPs, guarantees that the transitions in the target bit rates of consecutive IPs are smooth. The operations performed by the LT layer are as follows.

#### 3.1.1. Bit Rate Conformance Checking

Once the  $i$ th LT window has concluded (i.e. after encoding the  $i$ th IP), the total amount of bits produced by the encoded IPs in the window is accumulated as follows:

$$T_{LW}(i) = \sum_{\substack{l=i-N+1 \\ l>0}}^i T(l), \quad (1)$$

where  $T(l)$  is the total amount of bits employed for encoding the  $l$ th IP.

Subsequently,  $T_{LW}(i)$  is checked for deviations with respect to  $L_{th}(i)$  and  $U_{th}(i)$ , the lower and upper bit thresholds for the  $i$ th LT-window, respectively:

$$D(i) = \begin{cases} L_{th}(i) - T_{LW}(i) & , \text{ if } L_{th}(i) > T_{LW}(i) \\ U_{th}(i) - T_{LW}(i) & , \text{ if } U_{th}(i) < T_{LW}(i) \\ 0 & , \text{ otherwise,} \end{cases} \quad (2)$$

where  $D(i)$  represents the amount of bits out of the range of acceptance, which can be either positive or negative, depending on whether the bit count is below or exceeds the considered range, respectively. An in-depth description of how these thresholds evolve with  $i$  is provided in Subsection 3.1.3.

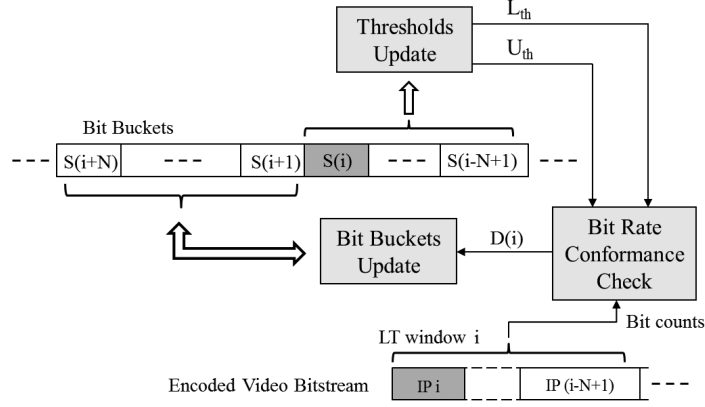


Figure 4: Block diagram of the LT layer.

### 3.1.2. Bit Bucket Update

As a way of smoothly compensating for the previous deviations, a bit bucket  $S(i)$  is defined for each IP. As mentioned, these buckets accumulate the positive or negative offsets to be applied to the target bit rate of each IP. At the beginning of the coding session, they are initialized to zero. The bit deviation  $D(i)$  observed after encoding the  $i$ th LT window is then distributed among the subsequent  $N$  bit buckets. Although other distributions are possible, in order to avoid coarse variations in the target bit rate from one IP to the next, an even distribution among the buckets has been adopted:

$$S(i+l) \leftarrow S(i+l) + \frac{D(i)}{N^2} \quad l=1, 2, \dots, N. \quad (3)$$

Fig. 5 illustrates the rationale behind Eq. (3) for the particular case of  $N=3$ . In this case, a certain deviation  $d$  under the target bit count is simulated in the  $i$ th IP, implying that this IP expends an amount of bits equal to the nominal target bits,  $B_{IP} = Mf^{-1}R_T$ , minus  $d$ , while the previous IPs expended exactly  $B_{IP}$  bits. Given that the window slides on an IP basis, this discrepancy affects only 3 consecutive LT windows, from the  $i$ th to the  $(i+2)$ th ( $D$  is zero for the rest of them), and each time it is detected, the contents of the following 3 buckets are updated in order to change accordingly the target bit rate of the next 3 IPs. Assuming that the coding of the following IPs exactly matches the modified target bit counts, namely  $T(i+l) = B_T + S(i+l)$  with  $l=1 \dots 2N-1$ , the total discrepancy  $d$  can be exactly compensated by introducing  $d/N^2 = d/9$  in the corresponding buckets. As can be seen, the variations between target bit counts for successive IPs are smooth, differing only in  $d/N^2$  bits from one IP to the next.

As an additional caution, the offset contained in each bit bucket  $S(i)$  must always be bounded by  $S_{max}$  to prevent the target bit counts of any of the IPs to reach beyond the limit established by the maximum bit rate  $R_M$ , and therefore we define:

$$S_{max} = \frac{M}{f} (R_M - R_T). \quad (4)$$

Specifically, if the application of Eq. (3) provides any  $S(i+l)$  greater than  $S_{max}$ , its value will be truncated to this bound and the remaining bits distributed among the rest of the buckets.

### 3.1.3. Lower- and Upper-Bound Threshold Update

Finally, the values of the lower- and upper-bound thresholds are also updated to be available for the bit rate conformance check of the next window, which will take place after encoding the following IP. Initially, and in order to avoid the under-use of bit resources, the lower threshold  $L_{th}(i)$  is set to the nominal amount



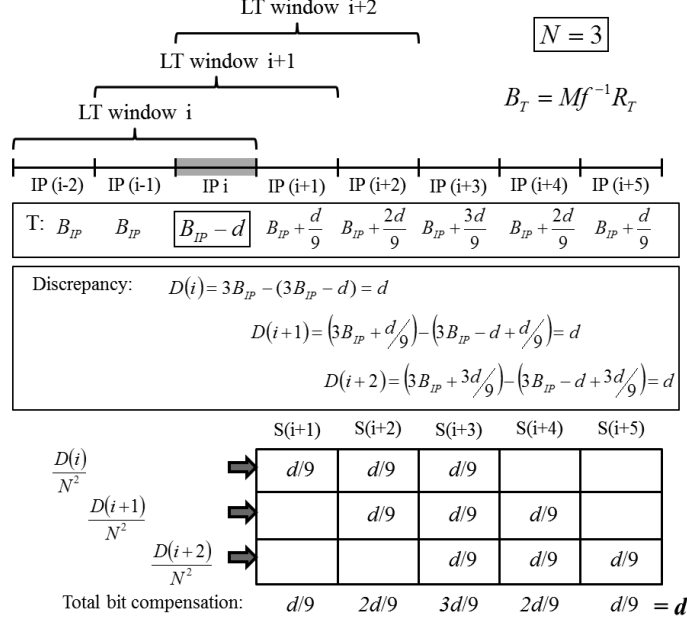


Figure 5: Bit buckets update example for  $N=3$ . All the IPs are supposed to exactly expend their bit budgets except for the  $i$ th IP, which presents a deviation of  $d$  bits.

of bits per LT window (that corresponding to a transmission at the target average bit rate). This default initial value, employed in the first window, is later updated to take into account the bit mismatch produced by previously encoded LT windows as:

$$L_{th}(i) = \frac{NM}{f} R_T + \sum_{\substack{l=i-N+1 \\ l>0}}^i S(l). \quad (5)$$

On the other hand, the upper-bound threshold  $U_{th}(i)$  is determined as a margin over  $L_{th}(i)$  according to the MEBC parameter. Again, this margin must be bounded by the maximum number of bits allowed per LT window, as follows:

$$U_{th}(i) = \min \left\{ \frac{NM}{f} R_M, \left( 1 + \frac{MEBC}{100} \right) L_{th}(i) \right\}, \quad (6)$$

where MEBC is expressed as a percentage of the target bit rate  $R_T$ .

### 3.2. Short-Term Layer

As outlined before, the task of the ST layer is to ensure that the target bit rate is met in a short-term period. To this end, instead of relying on complicated bit production models to obtain a proper quantization, the frame QP is simply increased or decreased anticipating the deviations of the bit rate with respect to the target. These deviations are predicted based on the recent history of the coding process, distinguishing between the different types of frames or levels of the hierarchical pattern.

For this purpose, a sliding ST window of  $M$  consecutive frames is defined as depicted in Fig. 6 where the  $j$ th ST window starts before encoding the  $j$ th frame and lasts until the  $(j+M-1)$ th frame in coding order. It is worth noticing that, in order to properly calculate the bit rate predictors as will be described in the following paragraphs,  $M$  should be a multiple of the number of frames in an IP. Since a large  $M$  produces

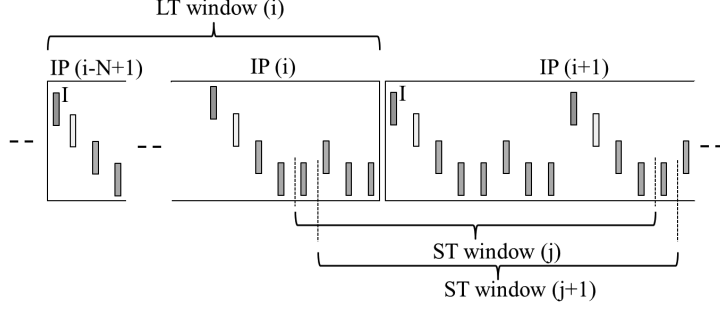


Figure 6: ST Window structure over frames in coding order.

a slow and, consequently, inaccurate adaptation,  $M$  has been selected as the number of frames in one IP in this paper.

Since the ST layer calculations only involve inter pictures, the bit budget or target bit count allocated for the single I frame present in the ST window,  $b_I(i)$ , will be considered identical to the size of the last encoded I picture, namely  $t_I(i-1)$ , and subtracted from the total bit budget of the window.

The complete process carried out at the ST layer is sketched in the block diagram in Fig. 7 comprising the following stages: *bit budget calculation*; *bit count prediction*; and *QP variation calculation*. The process carried out in these blocks is summarized in the next paragraphs following the *ST window definition and update* description.

### 3.2.1. ST Window Definition and Update

Let us assume that an IP consists of  $K$  hierarchical levels with  $M^{(k)}$  frames per level comprising  $\sum_k M^{(k)} = M$  frames. In this way, an IP structure with 16 frames (2 GoPs) like the one shown in Fig. 6 is split into a single I frame plus 1, 2, 4 and 8 Inter frames belonging to hierarchical levels 0 to 3, respectively. Since the length of a ST window is exactly the length of an IP, each ST window always covers the same number  $M^{(k)}$  of frames per hierarchical level, distributed among two consecutive IPs (see Fig. 6). Nevertheless, given that these two IPs may have different target bit rates (see Subsection 3.1.2), it is necessary to separately define the number of frames within the first and second IPs covered by the ST window.

To this end, we define  $m_i^{(k)}(j)$  as the number of Inter frames belonging to hierarchical level  $k$  that are spanned by the  $j$ th ST window and belong to the  $i$ th IP. By using this variable the total number of Inter frames in the  $i$ th IP covered by the  $j$ th ST window can be calculated as:

$$m_i(j) = \sum_{k=0}^{K-1} m_i^{(k)}(j). \quad (7)$$

For example, in Fig. 6, the  $j$ th window covers 4 Inter frames belonging to the  $i$ th IP (1 in layer  $k = 2$  and 3 in  $k = 3$ ).

At the beginning of the first ST window, all the frames covered belong to the first IP, i.e.:

$$\begin{aligned} m_i^{(k)}(j=1) &= \begin{cases} M^{(k)} & , k > 0 \\ M^{(k)} - 1 & , k = 0 \end{cases} \\ m_{i+1}^{(k)}(j=1) &= 0, \forall k. \end{aligned} \quad (8)$$

This initialization step is repeated every time a new IP begins but, when the ST slides a frame to the right (i.e.,  $j$  is increased by one unit) and an Inter frame of level  $\kappa$  is encoded, as shown in Fig. 6, the number of frames belonging to each IP needs to be updated by decreasing  $m_i^{(\kappa)}$  and increasing  $m_{i+1}^{(\kappa)}$  by one

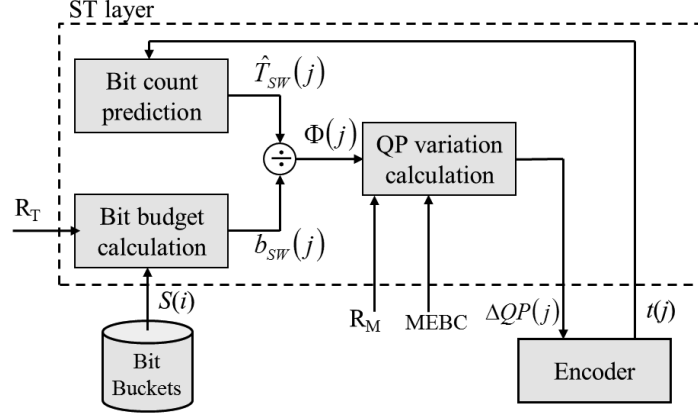


Figure 7: Block diagram of the ST layer.

unit:

$$\begin{aligned} m_i^{(\kappa)}(j) &= m_i^{(\kappa)}(j-1) - 1, \\ m_{i+1}^{(\kappa)}(j) &= m_{i+1}^{(\kappa)}(j-1) + 1. \end{aligned} \quad (9)$$

### 3.2.2. ST Window Bit Budget Calculation

This step links ST and LT layers by providing the bit budget for the current ST window. Each IP in the sequence may have a different bit budget considering the positive or negative offset stored in its corresponding bit bucket,  $S(i)$ . Excluding the bits reserved for encoding the I frame,  $b_I(i)$ , the average target bit count for the Inter frames in the  $i$ th IP can be calculated as follows:

$$b_B(i) = \frac{R_T f^{-1} M + S(i) - b_I(i)}{M - 1}. \quad (10)$$

The target bit rate for the complete ST window  $b_{SW}(j)$  is consequently calculated by weighting the values of  $b_B$  for both the  $i$ th and the  $(i+1)$ th IPs, according to the number of frames of each IP spanned by the window, as follows:

$$b_{SW}(j) = b_B(i)m_i(j) + b_B(i+1)m_{i+1}(j) + b_I(i). \quad (11)$$

### 3.2.3. ST Window Bit Count Prediction

Next, an estimation of the amount of bits that will be produced by the coding of the current  $j$ th ST window is calculated by combining a set of frame size predictors  $\hat{t}_B^{(k)}(j)$ , one per each hierarchical level. These predictors are calculated by exponentially averaging the sizes of every previously encoded frame belonging to the  $k$ th hierarchical level, for  $k = 0 \dots K - 1$ , as follows:

$$\hat{t}_B^{(k)}(j) = \sum_{\substack{m=1 \\ m \in \Omega^k}}^{j-1} \rho^{(j-m)} t_B^{(k)}(m), \quad (12)$$

where  $\Omega^k$  is the set of frame indexes of those Inter frames belonging to level  $k$ ;  $t_B^{(k)}(m)$  is the size in bits of Inter encoded frame  $m$  belonging to level  $k$ ; and the forgetting factor  $\rho$  equals 0.5.

As already described, two different IPs are spanned by the ST window and, although the set of predictors  $\hat{t}_B^{(k)}(j)$  may be accurate enough for those frames belonging to the first one (the  $i$ th IP), the produced bits

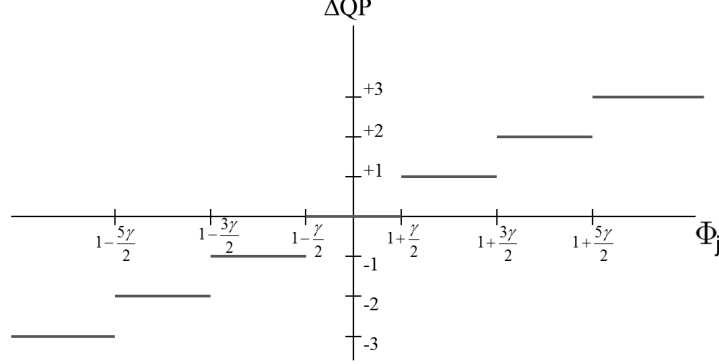


Figure 8: QP offset as a function of the risk parameter  $\Phi$ .

for the frames belonging to the  $(i+1)th$  IP may be notably different due to possible changes in the IP target bit rates carried out by the LT layer. On account of this difference, a *fitting* variable  $\mu(i)$  is defined:

$$\mu(i) = \frac{b_B(i+1)}{b_B(i)}. \quad (13)$$

Finally, the predictors are linearly combined by weighting their contributions by the number of frames per level in each IP spanned by the ST window and using the aforementioned *fitting* parameter, as follows:

$$\begin{aligned} \hat{T}_{SW}(j) = & b_I(i) + \sum_{k=0}^{K-1} m_i^{(k)}(j) \hat{t}_B^{(k)}(j) + \\ & \sum_{k=0}^{K-1} m_{i+1}^{(k)}(j) \mu(i) \hat{t}_B^{(k)}(j). \end{aligned} \quad (14)$$

#### 3.2.4. Quantization Parameter Variation

At this stage, the target bits obtained in Eq. (11) are compared to the expected bit count obtained by Eq. (14) for the entire ST window, obtaining a risk parameter:

$$\Phi_j = \frac{\hat{T}_{SW}(j)}{b_{SW}(j)}. \quad (15)$$

The base quantizer  $QP_0$  will be modified by means of a QP increment,  $\Delta QP(\Phi_j)$ , in the range  $\{-3, \dots, 3\}$ , depending on the risk parameter  $\Phi_j$  as depicted in Fig. 8. In order to comply with the restriction imposed by  $R_M$ , an additional check on the output bit rate is performed updating  $QP_0$  as follows:

$$QP_0(j) = QP_0(j-1) + \begin{cases} 3 & , \hat{T}_{SW} > \frac{MR_M}{f} \\ \Delta QP(\Phi_j) & , otherwise. \end{cases} \quad (16)$$

The step width  $\gamma$  employed for determining the  $\Delta QP$  in Fig. 8 represents the ST layer sensitivity to deviations in the bit rate. The smaller the  $\gamma$ , the faster the adaptation since the variations in the QP values are coarser, which could come in detriment of the overall subjective quality of the reconstructed sequence. Therefore, parameter  $\gamma$  should be properly selected in order to obtain a compromise between fast adaptation to the target bit rate and smooth variations of QP. As described in Section 2, when using hierarchical GoP structures, the QP can be increased from one temporal layer to the next in order to guarantee that the

distortion introduced by the quantization process affects the reconstructed frames according to the depth of each frame in the hierarchical tree, saving more bits for key frames ( $k=0$ ) and risking more distortion in those not used for reference ( $k=K-1$ ).

Therefore, the increment in Eq. (16) affects the corresponding value of  $QP_0$  independently of the QP cascading preset. Finally, the QP value incorporates the corresponding additional offset depending on the hierarchical level  $\kappa$  of the  $j$ th frame:

$$QP(j) = QP_0(j) + QP_{off}(\kappa), \quad (17)$$

where

$$QP_{off}(k) = \begin{cases} 0 & , if I \\ k+1 & , if Inter. \end{cases} \quad (18)$$

## 4. Experiments and Results

The proposed VBR control algorithm was implemented on the HM reference software version HM-11.0 [13]. Its performance was compared to those of other two methods: 1) constant QP cascading (CQPC) encoding, which can be seen as an unconstrained VBR control method [9] and was used as a reference for nearly constant quality video; and 2) the RCA proposed by *Li et al.* [29], which can be seen as reference for nearly constant bit rate video. Experiments concerning R-D performance, quality consistency and MEBC compliance have been carried out to compare the methods in terms of quality and applicability in a VBR scenario.

### 4.1. Experimental Setup

The evaluated systems were tested under the common conditions recommended in [31]. Additionally, considering that our proposal is designed to obtain a better performance on a long-term basis (as already explained in Section 3) two 5 minute long HD sequences at 25 fps, (*ElephantsDream\_1920x1080\_25* and *BigBuckBunny\_1920x1080\_25*) were tested in our experiments. These two sequences, available on-line in [32], are animated movies and represent a challenge from the rate control point of view because of their changes in video complexity and scene cuts.

Given the characteristics of the target encoding scenario, the dyadic hierarchical B-picture coding configuration was adopted due to its R-D benefits. Random Access (RA) Main and RA High Efficiency (Main10) coding settings were employed in our experiments. Class E video sequences were excluded from the experiments, following the recommendation in [31]. For the large video sequences, only RA Main coding settings were used, due to the extensive computation time required to encode large video files with the HM reference software.

The IP was selected according to the frame rate of each video sequence, being 16 frames for 20 fps, 24 for 24 fps, 32 for 30 fps, 48 for 50 fps, and 64 for 60 fps, as recommended in [31]. In this way, the intra refresh period is always around one second and, as will be described later, the LT window is around 10 second long.

To establish the set of target bit rates for the experiments, an encoding pass was first conducted using the default CQPC strategy in [13] with the following base QP values: 22, 27, 32, 37. These QP values were then used as InitialQP values and the resulting output bit rates as the target for both RCAs in order to provide a fair comparison.

The main RC encoding parameters employed are listed on Table 1. The ST window size  $M$  was set to one IP to allow the swift adaptation of QP to the changes in  $R_T$  (provided by the LT window) and, at the same time, to meet the constraint of containing always the same number of frames of each type and level. The values of MEBC and  $N$ , which represents the number of IPs covered by every LT layer window, were fixed experimentally<sup>1</sup> to  $N = 10$  and  $MEBC = 5\%$ , considering encoding performance, quality variation and behavior under VBR conditions.

<sup>1</sup>Results for these experiments can be found in <http://www.tsc.uc3m.es/~jlmolinero/Annex.pdf>

Table 1: RC encoding parameters.

Algorithm	Parameter	Value
Proposed VBR	$R_M$	$2 \times R_T$
	$MEBC$	5
	$\gamma$	1.875
	$N$	10
	$M$	1 IP
	$InitialQP$	22, 27, 32, 37
Li et al. [29]	$KeepHierarchicalBit$	1
	$LCULevelRateControl$	0
	$RCLCUSeparateModel$	1
	$RCForceIntraQP$	0
	$InitialQP$	22, 27, 32, 37

An appropriate value of  $\gamma$  was experimentally chosen considering the trade-off between target bit rate compliance and smooth quality evolution. An intensive validation experiment was conducted for the selection of  $\gamma$  as small variations produce significant changes on the performance of the algorithm. Thus,  $\gamma = 1.875$  was chosen according to our parametrization, but this value should be reconsidered if other RC parameters or window sizes were to be employed. Regarding the baseline RCA, the parameter values showed in Table 1 were used.

#### 4.2. Quality Consistency

The ability of our proposed VBR-RCA to provide a better quality consistency while maintaining similar *Bjontegard difference* (BD)-rate values with respect to the reference RCA will be assessed in this subsection. The quality consistency will be objectively measured in terms of the luma *peak signal-to-noise ratio* (PSNR) standard deviation ( $\sigma_{PSNR}$ ) along the whole video sequence.

Table 2 shows the quality consistency results. In particular, two measures are provided: average  $\sigma_{PSNR}$ , computed by averaging all the individual PSNR deviations from encodings of a particular class; and maximum  $\sigma_{PSNR}$ , that is, the maximum value of this set of individual measurements. As can be seen, our proposal presents a lower  $\sigma_{PSNR}$  than the reference RCA in all classes of video sequences, except for Class F, which will be discussed later. This lower quality variation comes from the limitation of the QP fluctuations. Furthermore, Figs. 9 and 10 graphically illustrate these results for the two long sequences. Fig. 9 illustrates, for the two assessed RCAs, representative luma PSNR and QP evolutions at medium-high bit rate, while Fig. 10 depicts the behavior at low bit rate. From these figures, we can conclude that our proposal produces more consistent and lower QP along the shown sequences. The same conclusion can be drawn in terms of PSNR, providing similar average values compared to the baseline but keeping a smoother time evolution. This behavior is specially remarkable when a scene change occurs, as can be observed in Fig. 9 for *Elephants\_Dream\_1920x1080\_25*, for example, around pictures #625 and #4200. Fig. 11 illustrates this fact in more detail by showing a zoom of zone around picture #6900, where the scene change makes the reference RCA to suddenly change the QP in order to adapt to the new scene. This causes a corresponding change in PSNR. However, our proposed VBR-RCA maintains a similar QP value between both scenes and adapts to the new situation smoothly, making the PSNR to evolve without sudden changes. In other words, compared to the baseline RCA, our proposal is able to produce larger bit rate fluctuations for the sake of PSNR consistency. This is the reason why we achieve significant lower  $\sigma_{PSNR}$  values in sequences with scene changes. To subjectively illustrate video quality consistency, Figs. 12 and 13 show two consecutive frames of the video sequence *BQSquare\_416x240\_60* extracted from an IP whose bit rate was 160.2 kbps for the reference RCA and 157.2 kbps for our VBR proposal, respectively. As can be seen in Fig. 12, which shows images produced by the baseline RCA, the water in the right image appears much better defined than in the left one, originating a flicker effect that can be subjectively annoying. On the other hand, in Fig. 13,

Table 2: Average/maximum  $\sigma_{PSNR}$  [dB] of the baseline RCA [29] and the proposed RCA for both RA Main and RA Main10 coding settings.

	Random Access Main		Random Access Main10	
	Reference (avg./max.)	Proposed (avg./max.)	Reference (avg./max.)	Proposed (avg./max.)
Class A	1.86 / 3.60	1.69 / 3.57	1.89 / 3.65	1.72 / 3.63
Class B	1.01 / 2.33	0.96 / 2.19	1.02 / 2.34	0.97 / 2.20
Class C	1.86 / 2.94	1.77 / 2.81	1.88 / 2.99	1.79 / 2.85
Class D	1.77 / 2.50	1.61 / 2.28	1.79 / 2.55	1.62 / 2.38
Class F	1.55 / 2.32	3.02 / 7.62	1.52 / 2.36	2.98 / 7.39
Elephant's Dream	21.37 / 21.68	9.22 / 9.35	- / -	- / -
Big Buck Bunny	13.24 / 13.54	5.79 / 5.92	- / -	- / -
<b>Average Except Class F</b>	<b>6.85 / 7.77</b>	<b>3.51 / 4.35</b>	<b>1.64 / 2.88</b>	<b>1.53 / 2.77</b>

which shows the same images produced by the proposed RCA, these changes are not so noticeable as the water texture seems to be quite similar in both frames. In any case, Figs. 12 and 13 should be taken as an illustration and the reader is referred to the encoded video sequences <sup>2</sup> to subjectively judge the quality consistency.

#### 4.3. R-D Performance

Tables 3 and 4 show the performance of both the proposed RCA and the baseline RCA with respect to CQPC encoding in terms of BD-rate and BD-PSNR, respectively, for different classes of video sequences. The *mean squared error* (MSE)-based sequence PSNR was used to compute the distortion part in both cases. This measure, which has already been included in the latest versions of HM [13], calculates first the average MSE for each video component (Y, U, V), and then combines the resulting  $MSE_Y$ ,  $MSE_U$  and  $MSE_V$  as follows:

$$MSE_{YUV} = \frac{4 \times MSE_Y + MSE_U + MSE_V}{6}, \quad (19)$$

and finally converts the result into PSNR to obtain the expected sequence BD-rate and BD-PSNR. From these results we can conclude that the smooth changes in the PSNR evolution provided by our proposal affect slightly the overall R-D performance incurring in moderate quality losses with respect to the baseline RCA. It is also worth mentioning that the poor BD-rate and BD-PSNR results achieved by both RCAs on both long HD sequences are due to the fact that our computation of the PSNR at a sequence level used an MSE-based averaging instead of a PSNR-based one (in dBs). Since the former averages square distortions, scenes with lower PSNR (i.e. with high video complexity) contribute more to the global results than if the latter averaging was used. For example, frames #2300 to #3100 of Elephant's Dream as illustrated in 9 significantly penalize the global PSNR. Besides, this effect is more demeaning on any RCA if compared to a constant QP encoding strategy since the latter usually produces (for a given target bit rate) higher quality on high complexity scenes and viceversa.

The results for Class F video sequences are significantly different from the rest. In particular, they are highly influenced by a particular video sequence, *SlideShow\_1280x720\_20*, where several transitions and scene changes take place in a short-term basis. This short-term behavior has a negative impact on our proposal's performance, which is clearly designed for optimizing its long-term performance and, consequently, cannot guarantee a good performance on a short-term basis for this specific type of sequences. Since video sequences exhibiting series of sudden content changes (happening from frame to frame) prevent our long-term RCA from adapting in a suitable time, an algorithm would have to be developed to detect such series of rapid changes. However, the development of this solution is out of the scope of this paper.

<sup>2</sup>The bitstream files from these video sequences can be found in: <http://www.tsc.uc3m.es/~jlmolinero/bitstreams.zip>

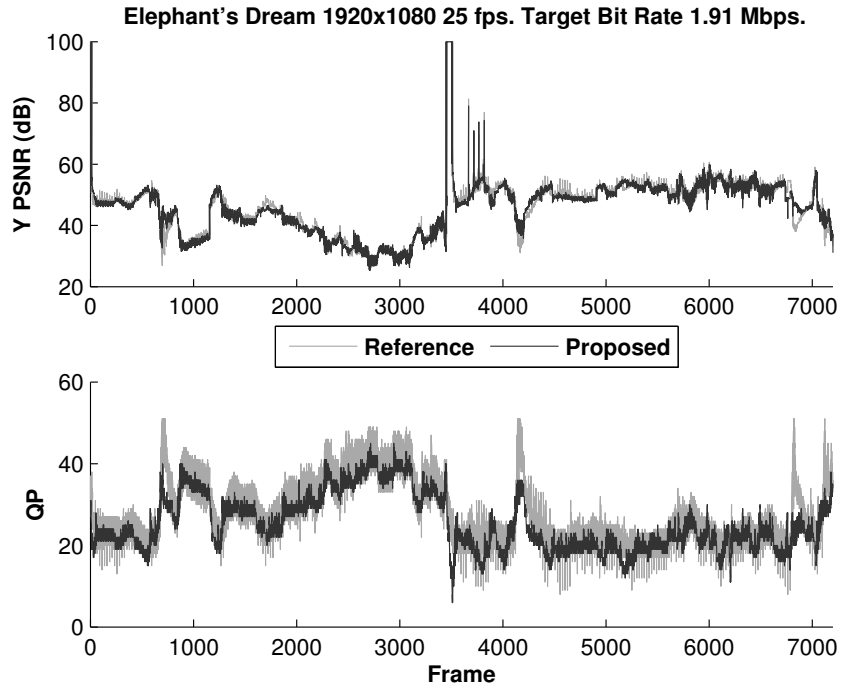


Figure 9: Luma PSNR and QP evolutions for *ElephantsDream\_1920x1080\_24* @ Target Bit Rate 1.91 Mbps.

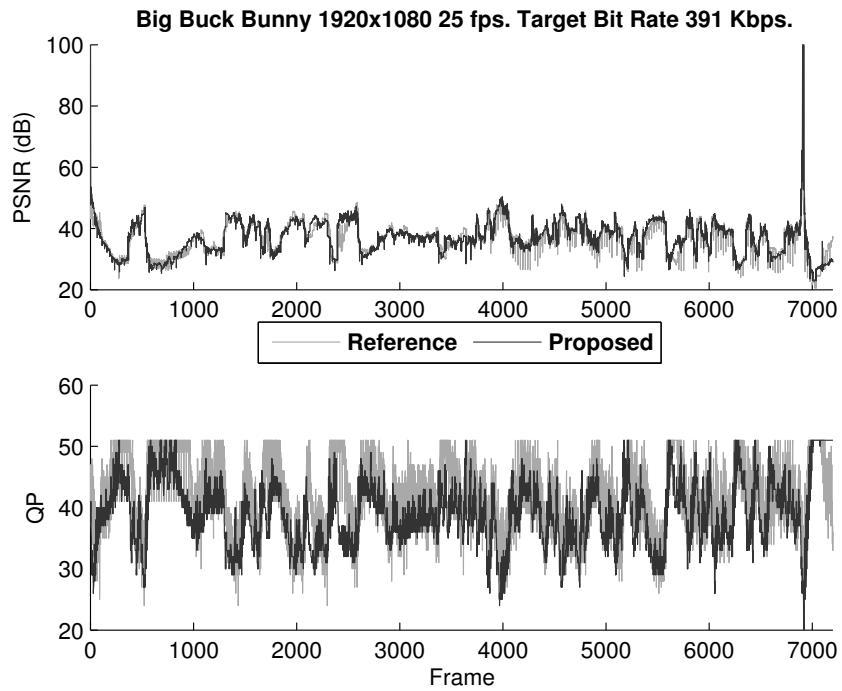


Figure 10: Luma PSNR and QP evolutions for *BigBuckBunny\_1920x1080\_24* @ Target Bit Rate 391 Kbps.



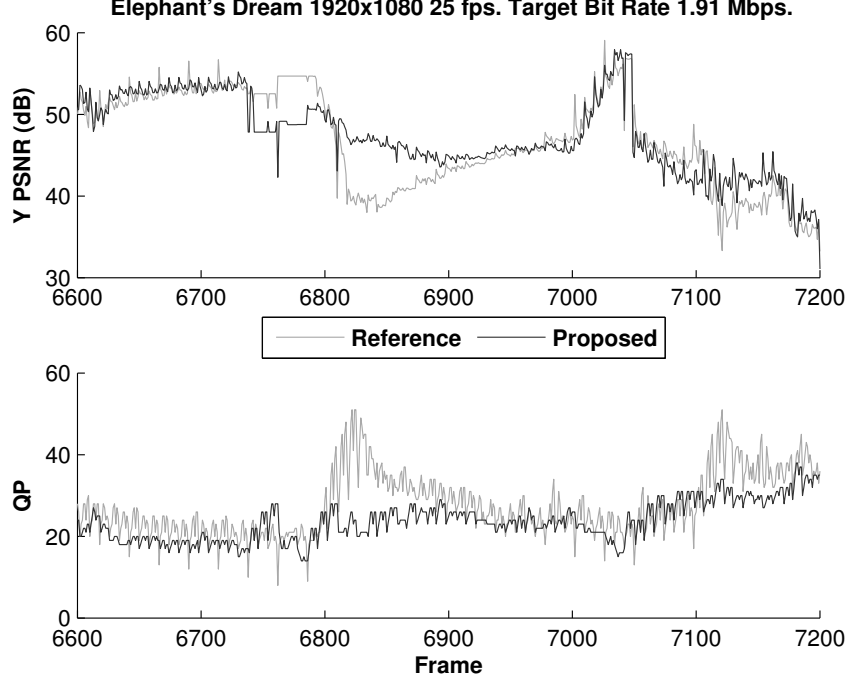


Figure 11: Luma PSNR and QP evolutions for *ElephantsDream\_1920x1080\_24* @ Target Bit Rate 1.91 Mbps around picture #6900.

#### 4.4. MEBC Compliance

In order to assess the suitability of an RCA for VBR applications,  $R_T$  and MEBC compliance should be evaluated. Thus, average and maximum bit rate errors for every video sequence class, calculated following

$$\text{Bit Rate Error} = \frac{R_{T_{RCA}} - R_T}{R_T} \times 100 \quad [\%], \quad (20)$$

where  $R_{T_{RCA}}$  is the output bit rate of the evaluated RCA, are presented in Table 5, for both RA Main and RA Main10 settings.

Considering the average bit rate error, the proposed scheme takes advantage of the extra bit allowance (bounded by MEBC), exhibiting higher deviations from  $R_T$  than the baseline RCA. This advantage allows it to keep a lower average QP during the whole coding process, as can be shown in Figs. 9 and 10. In fact, the average error only surpasses the established MEBC in sequence *BigBuckBunny\_1920x1080\_25* due to the abrupt scene change that takes place around frame #7000, which happens at the end of the sequence, and does not allow our proposal to properly converge again because of its long-term bit rate adjustment.

Class F video sequences also affect the behavior of the reference RCA, which produces higher bit rate error for this class when compared to the rest of the classes. However, as it is more suited for short-term transitions, the reference RCA adjusts faster and becomes less affected than the proposed scheme. These increments in the bit rate error come from video sequences where sudden content changes or scene changes take place as *NebutaFestival\_2560x1600\_60\_10bit\_crop* in Class A, and *SlideEditing\_1280x720\_30* and *SlideShow\_1280x720\_20* in Class F.

## 5. Conclusions and Further Work

In this paper, we have described a novel RCA for hierarchical video coding capable of taking advantage of the bit rate allowances of typical VBR applications such as video storage or VoD. Specifically, a two-level



Figure 12: Decoded frames #313 (left) and #314 (right) for *BQSquare\_416x240\_60* @ IP Bit Rate 160.2 Kbps using the baseline RCA.



Figure 13: Decoded frames #313 (left) and #314 (right) for *BQSquare\_416x240\_60* @ IP Bit Rate 157.2 Kbps using the proposed RCA.

VBR controller has been proposed and implemented on an HEVC encoder for our experiments. First, a long-term level of a few second duration is used for detecting the deviations from the target bit rate that are smoothly compensated using a smart bit allocation scheme. And second, a few-frame long short-term level is used for determining an adequate QP offset from the general QPC scheme.

Compared to the reference RCA described in [33], our proposal presents a smoother quality evolution in terms of PSNR deviation at the cost of a slight decrease in coding performance as measured by the BD-rate for both RA Main and RA Main10 settings. Its behavior is also adequate in terms of MEBC compliance (with the exception of some video sequences presenting sudden content changes from one frame to another that would require the use of a subsystem able to detect such situations). This makes our proposed algorithm more suitable under VBR environments compared with the reference RCA. This performance has been achieved with a minimum increase of computational complexity.

As future work, the following three improvements could be applied to the proposed algorithm:

- A scene change detection module should be helpful to solve the aforementioned problems of MEBC compliance. This module should be able to alert of sudden content changes and to provide the mechanisms to react promptly.
- While the proposed algorithm defines predictors for Inter frames depending on their depth in the hierarchical pattern, the solution adopted for predicting the size in bits of the next Intra frame to be encoded may be inaccurate for certain fast changing sequences (it simply relies on data from the last encoded Intra frame that could have nothing to do with the next one), and the algorithm could benefit from a more accurate prediction based, for example, in spatial complexity measures.

Table 3: R-D performance in terms of BD-rate (%) of the proposed RCA and the baseline RCA [29] both with respect to CQPC coding.

	Baseline RCA: $\Delta BD - R$ (CQPC)								Proposed RCA: $\Delta BD - R$ (CQPC)							
	Random Access Main				Random Access Main10				Random Access Main				Random Access Main10			
	Y	U	V	YUV	Y	U	V	YUV	Y	U	V	YUV	Y	U	V	YUV
Class A	13.0	2.2	2.2	12.8	13.0	1.1	0.7	12.8	13.7	11.1	11.5	13.6	13.6	9.9	10.3	13.5
Class B	3.0	2.9	0.3	2.9	2.6	2.3	-0.1	2.6	7.2	12.4	14.6	7.4	7.3	12.4	14.6	7.4
Class C	5.7	2.7	2.2	5.6	5.8	2.4	2.0	5.7	10.1	8.3	9.4	10.1	10.1	8.2	9.7	10.1
Class D	5.9	4.2	4.4	5.8	6.1	4.2	4.7	6.0	7.8	10.2	11.5	7.8	7.9	9.8	11.8	7.9
Class F	10.3	7.5	6.7	10.2	10.2	7.6	6.9	10.0	23.1	18.4	18.5	23.0	22.8	17.5	17.3	22.7
Elephant's Dream	118.0	-37.7	-27.2	116.4	-	-	-	-	122.4	-35.8	-27.6	120.6	-	-	-	-
Big Buck Bunny	96.4	46.8	42.1	94.6	-	-	-	-	106.1	63.1	63.4	104.5	-	-	-	-
<b>Overall</b>	<b>24.2</b>	<b>4.0</b>	<b>3.9</b>	<b>23.8</b>	<b>7.6</b>	<b>3.5</b>	<b>2.8</b>	<b>7.4</b>	<b>29.4</b>	<b>12.3</b>	<b>13.9</b>	<b>29.1</b>	<b>12.3</b>	<b>11.6</b>	<b>12.7</b>	<b>12.3</b>
<b>Average Except Class F</b>	<b>26.9</b>	<b>3.3</b>	<b>3.3</b>	<b>26.5</b>	<b>6.9</b>	<b>2.5</b>	<b>1.8</b>	<b>6.8</b>	<b>30.6</b>	<b>11.1</b>	<b>13.0</b>	<b>30.3</b>	<b>9.7</b>	<b>10.1</b>	<b>11.6</b>	<b>9.7</b>

Table 4: R-D performance in terms of BD-PSNR (dB) of the proposed RCA and the baseline RCA [29] both with respect to CQPC coding.

	Baseline RCA: $\Delta BD - PSNR$ (CQPC)								Proposed RCA: $\Delta BD - PSNR$ (CQPC)							
	Random Access Main				Random Access Main10				Random Access Main				Random Access Main10			
	Y	U	V	YUV	Y	U	V	YUV	Y	U	V	YUV	Y	U	V	YUV
Class A	-0.20	0.00	0.00	-0.15	-0.20	0.00	0.00	-0.18	-0.28	-0.15	-0.15	-0.28	-0.28	-0.14	-0.13	-0.27
Class B	-0.06	-0.04	0.00	-0.06	-0.06	-0.04	0.02	-0.06	-0.16	-0.18	-0.20	-0.16	-0.17	-0.16	-0.21	-0.17
Class C	-0.23	-0.05	-0.08	-0.20	-0.23	-0.08	-0.10	-0.23	-0.38	-0.18	-0.23	-0.35	-0.38	-0.20	-0.26	-0.38
Class D	-0.23	-0.08	-0.10	-0.20	-0.23	-0.10	-0.10	-0.23	-0.33	-0.20	-0.30	-0.33	-0.32	-0.22	-0.28	-0.32
Class F	-0.95	-0.48	-0.48	-0.90	-1.00	-0.53	-0.53	-0.98	-1.65	-0.83	-0.88	-1.60	-1.66	-0.81	-0.85	-1.61
Elephant's Dream	-3.00	1.00	0.80	-3.00	-	-	-	-	-3.00	1.00	0.80	-3.00	-	-	-	-
Big Buck Bunny	-3.10	-1.30	-1.00	-3.00	-	-	-	-	-3.10	-1.50	-1.30	-3.10	-	-	-	-
<b>Overall</b>	<b>-0.79</b>	<b>-0.13</b>	<b>-0.13</b>	<b>-0.75</b>	<b>-0.34</b>	<b>-0.15</b>	<b>-0.14</b>	<b>-0.33</b>	<b>-0.97</b>	<b>-0.30</b>	<b>-0.33</b>	<b>-0.96</b>	<b>-0.56</b>	<b>-0.31</b>	<b>-0.35</b>	<b>-0.55</b>
<b>Average Except Class F</b>	<b>-0.75</b>	<b>-0.06</b>	<b>-0.06</b>	<b>-0.72</b>	<b>-0.18</b>	<b>-0.05</b>	<b>-0.05</b>	<b>-0.17</b>	<b>-0.84</b>	<b>-0.19</b>	<b>-0.23</b>	<b>-0.83</b>	<b>-0.29</b>	<b>-0.18</b>	<b>-0.22</b>	<b>-0.29</b>

Table 5: Average/maximum bit rate error [%] of the baseline RCA [29] and the proposed RCA for both RA Main and RA Main10 coding settings.

	Random Access Main		Random Access Main10	
	Reference	Proposed	Reference	Proposed
	(avg./max.)	(avg./max.)	(avg./max.)	(avg./max.)
Class A	0.43 / 1.67	2.68 / 5.72	0.50 / 1.96	2.58 / 5.76
Class B	0.26 / 2.32	1.59 / 4.62	0.26 / 2.98	1.32 / 4.52
Class C	0.12 / 0.91	0.90 / 2.75	0.11 / 0.94	0.91 / 2.28
Class D	0.42 / 2.52	1.79 / 3.90	0.36 / 2.64	1.76 / 3.20
Class F	2.59 / 8.04	5.21 / 12.72	6.85 / 49.53	7.93 / 48.56
Elephant's Dream	0.48 / 1.47	3.05 / 3.65	- / -	- / -
Big Buck Bunny	0.04 / 0.10	6.09 / 6.32	- / -	- / -
<b>Average Except Class F</b>	<b>0.29 / 1.50</b>	<b>2.68 / 4.49</b>	<b>0.31 / 2.13</b>	<b>1.64 / 3.94</b>

- Finally, although the value selected for  $\gamma$  in our experiments seems to be a good trade-out between output bit rate and quality variations, preliminary experiments have demonstrated that there could be a relationship between the sequence contents and the best  $\gamma$ , that could be further exploited.

## Acknowledgment

This work has been partially supported by the National Grant TEC2011-26807 of the Spanish Ministry of Economy and Competitiveness.

## References

- [1] ISO/IEC, Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video, ITU-T Recommendation H.262-ISO/IEC 13818-2, MPEG-2.
- [2] ITU-T, Video Coding for Low Bitrate Communication, ITU-T Draft Recommendation H.263 Version 1.
- [3] T. Sikora, The MPEG-4 Video Standard Verification Model, *Circuits and Systems for Video Technology*, IEEE Transactions on 7 (1) (1997) 19–31.
- [4] Advanced Video Coding for Generic Audiovisual Services: ITU-T Recommendation H.264 - International Standard ISO/IEC 14496-10 (Feb. 2014).  
URL <http://handle.itu.int/11.1002/1000/12063>
- [5] High Efficiency Video Coding: Recommendation ITU-T H.265 - International Standard ISO/IEC 23008-2 (Oct. 2014).  
URL <http://handle.itu.int/11.1002/1000/12296>
- [6] G. Sullivan, J. Ohm, W. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) Standard, *Circuits and Systems for Video Technology*, IEEE Transactions on 22 (12) (2012) 1649–1668.
- [7] H. Schwarz, D. Marpe, T. Wiegand, Analysis of Hierarchical B Pictures and MCTF, in: *Multimedia and Expo*, 2006 IEEE International Conference on, 2006, pp. 1929–1932.
- [8] W. Wan, Y. Chen, Y.-K. Wang, M. Hannuksela, H. Li, M. Gabbouj, Efficient Hierarchical Inter Picture Coding for H.264/AVC Baseline Profile, in: *Picture Coding Symposium*, 2009. PCS 2009, 2009, pp. 1–4.
- [9] T. Lakshman, A. Ortega, A. Reibman, VBR Video: Tradeoffs and Potentials, *Proceedings of the IEEE* 86 (5) (1998) 952–973.
- [10] A. Ortega, Variable Bit-Rate Video Coding, *Compressed Video over Networks*, M.-T. Sun and A. R. Reibman, Eds. New York: Marcel Dekker (2000) 343–382.
- [11] M. Wien, H. Schwarz, T. Oelbaum, Performance Analysis of SVC, *Circuits and Systems for Video Technology*, IEEE Transactions on 17 (9) (2007) 1194–1203.
- [12] B. Li, D. Zhang, H. Li, J. Xu, QP Determination by Lambda Value, *JCTVC-I0426*, 9th JCTVC Meeting.
- [13] HEVC software repository - Revision 4263: /tags.  
URL [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/tags](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags)
- [14] Y. Yokoyama, Y. Ooi, A Scene-Adaptive One-Pass Variable Bit Rate Video Coding Method for Storage Media, in: *Image Processing*, 1999. ICIP 99. Proceedings. 1999 International Conference on, Vol. 3, 1999, pp. 827–831.
- [15] Z. Chen, K. N. Ngan, *Recent Advances in Rate Control for Video Coding*, Vol. 22, Elsevier Science Inc., New York, NY, USA, 2007, pp. 19–38.
- [16] S. Sanz-Rodríguez, F. Díaz-de María, In-Layer Multibuffer Framework for Rate-Controlled Scalable Video Coding, *Circuits and Systems for Video Technology*, IEEE Transactions on 22 (8) (2012) 1199–1212.
- [17] M. Naccari, F. Pereira, Quadratic Modeling Rate Control in the Emerging HEVC Standard, in: *Picture Coding Symposium (PCS)*, 2012, 2012, pp. 401–404.
- [18] H. Choi, J. Nam, J. Yoo, D. Sim, I. Bajić, Rate Control Based on Unified RQ model for HEVC, *JCTVC-H0213*, 8th JCTVC Meeting.
- [19] S. Ma, Z. Li, F. We, Proposed Draft of Adaptive Rate Control, *JVT-H017*, 8th JVT Meeting.
- [20] J. Si, S. Ma, W. Gao, M. Yang, Adaptive Rate Control for HEVC, *JCTVC-J0057*, 10th JCTVC Meeting.
- [21] S. Ma, W. Gao, Y. Lu, Rate-Distortion Analysis for H.264/AVC Video Coding and its Application to Rate Control, *Circuits and Systems for Video Technology*, IEEE Transactions on 15 (12) (2005) 1533–1544.
- [22] S. Wang, S. Ma, S. Wang, D. Zhao, W. Gao, Rate-GOP Based Rate Control for High Efficiency Video Coding, *Selected Topics in Signal Processing*, IEEE Journal of 7 (6) (2013) 1101–1111.
- [23] Z. He, Y. K. Kim, S. Mitra, Low-delay Rate Control for DCT Video Coding via  $\rho$ -domain Source Modeling, *Circuits and Systems for Video Technology*, IEEE Transactions on 11 (8) (2001) 928–940.
- [24] Y.-J. Yoon, H. Kim, S. heung Jung, D. Jun, Y. Kim, J. S. Choi, S.-J. Ko, A New Rate Control Method for Hierarchical Video Coding in HEVC, in: *Broadband Multimedia Systems and Broadcasting (BMSB)*, 2012 IEEE International Symposium on, 2012, pp. 1–4.
- [25] S. Sanz-Rodríguez, T. Schierl, A Rate Control Algorithm for HEVC with Hierarchical GOP Structures, in: *Acoustics, Speech, and Signal Processing*, 2013 IEEE International Conference on, 2013.
- [26] X. Liang, Q. Wang, Y. Zhou, B. Luo, A. Men, A Novel R-Q Model Based Rate Control Scheme in HEVC, in: *Visual Communications and Image Processing (VCIP)*, 2013, 2013, pp. 1–6.
- [27] L. Sun, O. Au, C. Zhao, F. Huang, Rate Distortion Modeling and Adaptive Rate Control Scheme for High Efficiency Video Coding (HEVC), in: *Circuits and Systems (ISCAS)*, 2014 IEEE International Symposium on, 2014, pp. 1933–1936.
- [28] N. Kamaci, Y. Altunbasak, R. Mersereau, Frame Bit Allocation for the H.264/AVC Video Coder via Cauchy-density-based Rate and Distortion Models, *Circuits and Systems for Video Technology*, IEEE Transactions on 15 (8) (2005) 994–1006.
- [29] B. Li, H. Li, L. Li, J. Zhang, (Lambda) Domain Rate Control Algorithm for High Efficiency Video Coding, *Image Processing*, IEEE Transactions on 23 (9) (2014) 3841–3854.

- [30] M. Rezaei, M. Hannuksela, M. Gabbouj, Semi-fuzzy rate controller for variable bit rate video, *Circuits and Systems for Video Technology*, IEEE Transactions on 18 (5) (2008) 633–645.
- [31] F. Bossen, Common Test Conditions and Software Reference Configurations, JCT-VC 12th Meeting, JCTVC-L1100.
- [32] Xiph.org Video Test Media [Dorf's Collection].  
URL <https://media.xiph.org/video/derf/>
- [33] B. Li, H. Li, L. Li, J. Zhang, Rate Control by R-Lambda Model for HEVC, JCTVC-K0103, 11th JCTVC Meeting.